



DEPARTMENT OF MATHEMATICS AND COMPUTER & INFORMATION SCIENCE

SYLLABUS

COMPUTER PROGRAMMING I CS2510

Prerequisite: MA 1020 or equivalent.

COURSE DESCRIPTION: CS 2510 is an introduction to procedural and object-oriented programming methodology using the Java programming language. Java is among the most in-demand programming languages, mainly because of how versatile and compatible it is. This course emphasizes engaging students in Computer Science through a hands-on programming approach to learning the fundamentals. That means instead of just introducing principles to students with no initial context, principles are introduced in the context of problem-solving in class. Topics include variables, data types, operators, expressions, the flow of control, classes, objects, methods, i/o operations, debugging, and testing. It includes programming projects of an interdisciplinary nature and is suitable as an introductory programming course for all majors. Student participation in the online Lab is required.

COURSE OBJECTIVES: This course introduces the student to fundamental object-oriented computer programming concepts and facilitates the development of the necessary computer programming skills to a level where they are prepared for CS2511 Computer Programming II. Students will continue to use these skills in all of the courses in the computer science track.

PROGRAM-LEVEL STUDENT LEARNING OUTCOMES (PSLO):

1. Problem Solving through iterative design and algorithmic thinking including
 - A. Problem Formulation through Abstraction.
 - B. Expressing Solutions Computationally.

TEXTBOOK: **Java Illuminated: An Active Learning Approach** (5th Edition), by Julie Anderson and Herve Franceschi, published by Pearson, ISBN: 978-1-284-14099-6

In this course, Chapters 1-6 are covered. Please come to class ready to take notes and brainstorm to solve problems using the Java Programming Language.

TOPICS TO BE COVERED *

1. Chapter 1 Introduction to Programming and the Java Language

- 1.1 Basic Computer Concepts
 - 1.1.1 Hardware
 - 1.1.2 Operating Systems
 - 1.1.3 Application Software
 - 1.1.4 Computer Networks and the Internet

1.3 Data Representation

1.3.1 Binary Number

1.3.2 Using Hexadecimal Numbers to Represent Binary Numbers

1.3.3 Representing Characters with the Unicode Character Set

1.4 Programming Languages

1.4.1 High- and Low-Level Languages

1.4.2 An Introduction to Object-Oriented Programming

1.4.3 The Java Language

1.5 An Introduction to Programming

1.5.1 Programming Basics

1.5.2 Program Design with Pseudocode

1.5.3 Developing a Java Application

1.5.4 Writing a First Java Application

2. Chapter 2 Programming Building Blocks – Java Basics

2.1 Java Application Structure

2.2 Data Types, Variables, and Constants

2.2.1 Declaring Variables

2.2.2 Integer Data Types

2.2.3 Floating-Point Data Types

2.2.4 Character Data Type

2.2.5 Boolean Data Type

2.2.6 The Assignment Operator, Initial Values, and Literals

2.2.7 *String* Literals and Escape Sequences

2.2.8 Constants

2.3 Expressions and Arithmetic Operators

2.3.1 The Assignment Operator and Expressions

2.3.2 Arithmetic Operators

2.3.3 Operator Precedence

2.3.5 Integer Division and Modulus

2.3.6 Division by Zero

2.3.7 Mixed-Type Arithmetic and Type Casting

2.3.8 Shortcut Operators

3. Chapter 3 Object-Oriented Programming Part 1: Using Classes

3.1 Class Basics and Benefits

3.2 Creating Objects Using Constructors

3.3 Calling Methods

3.4 Using Object References

3.6 The Java Class Library

3.7 The *String* class

3.8 Formatting Output with the *DecimalFormat* Class

3.9 Generating Random Numbers with the *Random* Class

3.10 Input from the Console Using the *Scanner* Class

3.11 Calling *Static* Methods and Using *Static* Class Variables

3.12 Using *System.in* and *System.out*

3.13 The *Math* Class

3.14 Formatting Output with the *NumberFormat* Class

3.15 The *Integer*, *Double*, *Character*, and Other Wrapper Classes

4. Chapter 4 Introduction to Graphical Applications

- 4.1 JavaFX Application Structure
- 4.2 The Graphics Coordinate System and Color
- 4.3 Drawing Shapes and Text
- 4.4 Drawing Custom Shapes

5. Chapter 5 Flow of Control, Part 1: Selection

- 5.1 Forming Conditions
 - 5.1.1 Equality Operators
 - 5.1.2 Relational Operators
 - 5.1.3 Logical Operators
 - DeMorgan's Laws
- 5.2 Simple Selection with *if*
- 5.3 Selection Using *if/else*
- 5.4 Selection Using *if/else if*
- 5.5 Sequential and Nested *if/else* Statements
- 5.6 Testing Techniques for *if/else* Statements
- 5.8 Comparing Floating-Point Numbers
- 5.9 Comparing Objects
 - 5.9.1 The *equals* Method
 - 5.9.2 *String* Comparison Methods
- 5.10 The Conditional Operator (?:)
- 5.11 The *switch* Statement

6. Chapter 6 Flow of Control, Part 2: Looping

- 6.1 Event-Controlled Loops Using *while*
- 6.2 General Form for *while* Loops
- 6.3 Event-Controlled Looping
 - 6.3.1 Reading Data from the User
 - 6.3.2 Reading Data from a Text File
- 6.4 Looping Techniques
 - 6.4.1 Accumulation
 - 6.4.2 Counting Items
 - 6.4.3 Calculating an Average
 - 6.4.4 Finding Maximum or Minimum Values
- 6.5 Type-Safe Input Using *Scanner*
- 6.6 Constructing Loop Conditions
- 6.7 Testing Techniques for *while* Loops
- 6.8 Event-Controlled Loops Using *do/while*
- 6.10 Count-Controlled Loops Using *for*
 - 6.10.1 Basic Structure of *for* Loops
 - 6.10.2 Constructing *for* Loops
 - 6.10.3 Testing Techniques for *for* Loops
- 6.11 Nested Loops

* The topics may vary slightly and need to be adjusted as we move through the semester.